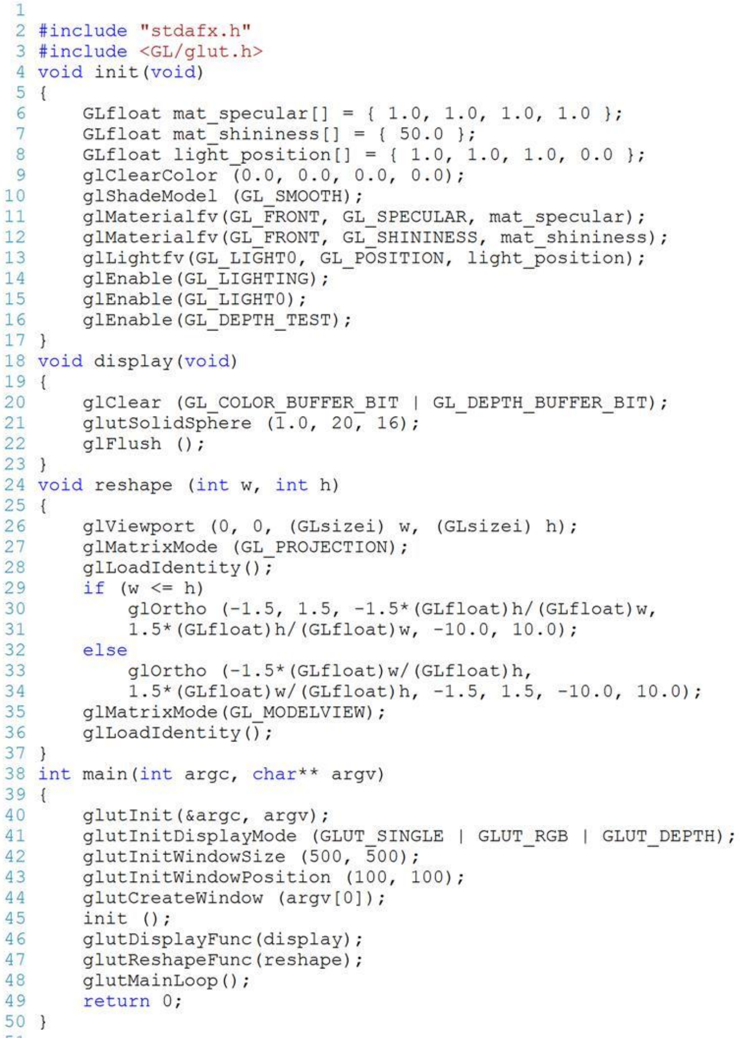
Tanta university
Faculty of engineering
Computer and automatic control department

Computer graphics course
Second year students
Sheet 1, Date : 15/02/2012

Setting up OpenGL

1. You should discuss with a demonstration the following issues
    - The installation of a C/C++ compiler with the required SDKs
    - OpenGL graphics library
    - OpenGl three parts: gl, glu, and glut
    - Setting up the OpenGL on your machine
    - Compile and run the program below (The program is just given as a verification tool for testing the installation, we did not study yet the contents of the program)



```

#include "stdafx.h"
#include <GL/glut.h>
void init(void)
{
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat mat_shininess[] = { 50.0 };
    GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_SMOOTH);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}
void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glutSolidSphere (1.0, 20, 16);
    glFlush ();
}
void reshape (int w, int h)
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    if (w <= h)
        glOrtho (-1.5, 1.5, -1.5*(GLfloat)h/(GLfloat)w,
        1.5*(GLfloat)h/(GLfloat)w, -10.0, 10.0);
    else
        glOrtho (-1.5*(GLfloat)w/(GLfloat)h,
        1.5*(GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return 0;
}
```

2. Computer graphics is concerning with all aspects of producing pictures or images using a computer. It specifically deals with images/pictures produced using programs
    - Writing a program in a general purpose language to produce an image

- Using a graphics software package to draw an image

We exclude that act of acquiring images using cameras (or other sensors) and manipulating it using the computer as this enter in the field of digital image processing rather that computer graphics

3.
1. Display of information
2. The human visual system is unrivaled as a pattern recognizer. We can employ computer graphics with that fact to convey information to human in learning, illustrations and presentations of materials so aiding viewers in understanding information (Information visualization)
3. Design(CADs systems)
4. Simulation and modeling (ex. Graphical flight simulator: real time graphics production, games, VR)
5. User interfaces

4. Images are better displayed than text on CRT monitors. By nature, the illumination of the phosphor dots on the CRT monitors change gradually from pixels to the adjacent pixel compared to the somewhat abrupt change on LCD monitors. This causes the gradual changes in color between the image pixels that gives the required smoothness of the picture. The same characteristic make the text appears more better on the LCD monitors since the abrupt changes makes the text clear. This in addition to some other factors like the dependency of the view quality on the viewer angle on LCD and high illumination on CRT monitors

5.
1. In two dimensions utilities/libraries, Forming image is done using the simple two dimensional geometrical entities (line, points, polygons) the libraries usually contains two types of functions:
    a. Low level functions to rasterizing the 2D entities in FB
    b. High-level function to form three dimensions object images using 2D primitives

    In such applications/libraries, it's the responsibility of graphics creator to form three dimensions images using the two dimensions primitives
2. In #D graphics utilities/libraries, Generating images is done through using a model for image generation that imitates optical imaging systems (cameras and human visual system)
    a. Specifying what exists in the scene, where the light sources(s) is(are) located, what is the nature of the scene materials, etc.
    b. A special software (graphics library) and hardware (GPU) cooperates to produce the scene according to your specification by applying the imaging model